Theory Reasoning with Answer Set Programming

Sebastian Schellhorn[1]

[1]University of Potsdam, Germany

Abstract. Answer Set Programming (ASP) is a well known declarative solving paradigm to model and solve efficiently combinatorial real world problems. But it may suffer from grounding bottleneck by represent and treat elements of other theories (eg linear constraints) or variables over multivalued domains. Addressed to this problem, my research focus on ASP modulo theory reasoning. The recent available system *clingo* 5 provides a generic interface to enhance ASP with theory reasoning capabilities. I instantiate this framework with linear constraints and elaborate upon its formal properties. My goal is to extend the picture ASP modulo theories and its relations. Furthermore, my research interests focus on extensions of stable model semantics regarding foundedness regarding multivalued domains. Finally, I show some state of the art approaches addressed to these problems, main issues and ideas to tackle them.

1 Introduction

Answer Set Programming (ASP[6]) is a well known declarative problem solving paradigm to efficiently solve combinatorial problems. It features a simple and flexible modeling language. Thus with ASP it is possible to describe real world problems in an intuitive way. State-of-the-art ASP solvers use high performance solving technologies and are able to efficiently solve several of these real world problems. However, certain problems require a more natural modeling with constraints over reals or integers. In this case, ASP solvers suffer from the grounding bottleneck by represent and treat elements of other theories invoke non-Boolean constraints (eg linear constraints) or variables over, probably infinite, multivalued domains. Addressed to this problem, my research focus on ASP modulo theory reasoning.

The recent available system *clingo* 5 [3] provides a generic interface to enhance ASP with theory reasoning capabilities. I instantiate this framework with Linear Programming (LP[1]) constraints and elaborate upon its formal properties. My goal is to extend and generalize this elaboration on formal relations of ASP with different theories, solve corresponding issues and extend the picture of ASP semantics and theory reasoning.

Furthermore, my research interests focus on extensions of stable model semantics regarding foundedness over multivalued domains. In particular, I am interested in stable model semantics extensions based on logic of Here-and-There

(HT[11]) regarding to default values of multivalued domain variables, foundedness, partial functions and aggregates. Thus I plan to achieve a Bound Founded ASP (BFASP[5]) semantics for arbitrary ordered multivalued variable domains.

Finally, in the following I show some state-of-the-art approaches addressed to these problems, clarify main issues and present ideas to tackle them.

2 ASP modulo Linear Constraints

In the past there were several approaches to combine ASP with constraints starting with [12]. Usual approaches are satisfiability checks on full assignments by using dedicated solver as a black box or adapting the underlying ASP solving algorithms to achieve an extension to a desired theory. With the recent *clingo* 5 interface, following the approach of lazy theory solving[4], it is possible to extend ASP with theories in a much easier way than before. This interface offers the opportunity to come up with an own propagator for a specific theory and use it during search on partial assignments to derive new information given by the theory or rule out search trees which are inconsistent with the theory.

I first started to study linear constraints over reals and its possible treatments, like translation based approaches, which still suffer from the grounding bottleneck. Afterward, I combined ASP with linear constraints over reals and integers on a system and theoretical level using the mentioned interface. Thereby, I abstract from the specific semantics of the theory by considering the linear constraint atoms (lc-atoms) associated with its constraints, and deal with the constraints as usual in LP. My *clingo* derivative *clingo*[LP], uses an easy modifiable Python script, providing a simple propagator. The propagator follows a state based approach. To solve the sub problem given by a set of linear constraints *clingo*[LP] supports LP solvers *cplex* and *lpsolve* as black boxes, which are plugged by its Python interface. Thus, to the best of my knowledge, it is the first time that Mixed Integer Linear Programming (MILP) solvers are combined with ASP without using a translation based approach for example like *mingo* [13] does. *clingo*[LP] checks during search if a corresponding set of linear constraints is satisfiable. Mention that it is the first implementation of *clingo* with linear constraints over reals. Furthermore, it achieves superior results on metabolic network completion of *Escherichia coli*[10], by using a new hybrid approach that integrates a previous qualitative ASP approach with quantitative means of a linear constraint set.

On the theoretical side, there are in general four possibilities to interpret lc-atoms. I analyzed them regarding their lc-stable models and compared them to each other and regular stable models to elaborate their formal relations. I figured out that most of them can be ordered independent of the corresponding theory, except for one proposition. Furthermore, I found translations to represent the two incomparable but most intuitive possibilities in sense of ASP by each other regarding to programs with specific properties.

Finally, with *clingo*[LP] I finished one practical and theoretical part of my research. An open issue would be to generalize and elaborate these propositions

and translations to less restrictive conditions and other theories. Additionally in a long run, I want to extend *clingo*[LP] by a kind of domain propagation over reals to avoid the intermediate search regarding to each partial assignment given by the black box approach.

3 ASP Foundedness over Multivalued Domains

On the other hand, my research focuses on derivatives of ASP semantics combined with default values over multivalued domains, foundedness, partial functions and aggregates.

The foundedness idea of BFASP represented in [5] is to derive a maximal value of a multivalued domain variable where we have a reason for. To achieve foundedness for variables with arbitrary ordered multivalued domains it is necessary to introduce a monotonic operator to declare reasonable values for assignments. Since the presented semantics of BFASP lacks the intuition of ASP in some cases, there is the need to take up the idea and reinvent it in different settings like logic of Here-and-There.

So far, there exists several ideas to achieve default valuations for multivalued domains like the approach of extending the logic of Here-and-There wich constraints [8], which allows default ranges for assignments. But this formalism itself is not able to grasp the intuition of BFASP, since their assignment operator uses an anti-monotonic relation. Furthermore, this approach misses an order on assignments to express preferred valuations.

Thus, I plan to come up with new stable model semantics for linear constraint variables with arbitrary ordered multivalued domains closely related to BFASP and logic of Here-and-There with constraints. To this end, I figured out two possible semantics in the context of logic of Here-and-There to achieve this idea of BFASP, which are closely related to ASP semantics.

While thinking about new semantics, I recognize similar problems as we can observe with aggregates and partial functions regarding to monotonic and anti-monotonic behaviors. Thus, I want to find relations of my semantics to aggregates and partial functions to extend the picture of their treatments and investigate reasons for counter intuitive behaviors as well. To this end, I reformulated different aggregate semantics in context of logic of Here-and-There to set the same footing for my comparison. Finally, I want to figure out the strength and weakness of my semantics compared to well known approaches.

However, in my main research tasks I focused on linear constraints over real-valued variables, but it is also possible to think about arbitrary functions over ordered multivalued domains and their monotonic behaviors to extend this approach.

References

1. G. Dantzig. *Linear Programming and Extensions.* Princeton, 1963.

2. M. Carro and A. King, editors. *Technical Communications of ICLP'16*, OASIcs, 2016.
3. M. Gebser, R. Kaminski, B. Kaufmann, M. Ostrowski, T. Schaub, and P. Wanko. Theory solving made easy with clingo 5. In [2], 2:1-2:15.
4. C. Barrett, R. Sebastiani, S. Seshia, and C. Tinelli. Satisfiability modulo theories. In *Handbook of Satisfiability*, 26:825-885. IOS, 2009.
5. R. Aziz. Bound Founded Answer Set Programming. In CoRR, abs/1405.3367, 2014.
6. M. Gebser, R. Kaminski, B. Kaufmann and T. Schaub. *Answer Set Solving in Practice*, Morgan and Claypool Publishers, 2012.
7. R. Kambhampati, editor. *Proceedings of IJCAI'16*, IJCAI/AAAI Press, 2016.
8. P. Cabalar, R. Kaminski, M. Ostrowski and T. Schaub. An ASP Semantics for Default Reasoning with Constraints. In [7], 1015-1021, 2016.
9. M. Balduccini and T. Janhunen, editors. *Proceedings of LPNMR'17*, Springer, 2017.
10. C. Frioux and T. Schaub and S. Schellhorn and A. Siegel and P. Wanko. Hybrid Metabolic Network Completion. In [9], to appear, 2017.
11. D. Pearce. A new logical characterisation of stable models and answer sets. In *Proceedings of the Sixth International Workshop on Non-Monotonic Extensions of Logic Programming (NMELP'96)*, pages 57–70. Springer-Verlag, 1997.
12. S. Baselice, P. Bonatti and M. Gelfond. Towards an integration of answer set and constraint solving. In *Proceedings of the Twenty-first International Conference on Logic Programming (ICLP'05)*, M. Gabbrielli and G. Gupta, Eds. Lecture Notes in Computer Science, vol. 3668. Springer-Verlag, 52–66, 2005.
13. G. Liu, T. Janhunen, and I. Niemelä. Answer set programming via mixed integer programming. In *Proceedings of KR'12*, 32-42. AAAI, 2012.